Word Prediction Using LSTM Model

Abdullah Masood Mughal.

FAST NUCES Islamabad Pakistan

i210822@nu.edu.pk

***Abstract*—This research develops an LSTM-based model for next-word prediction using Shakespeare's plays. The dataset comprises 111,396 rows, with 3,000 selected for training. Through a series of preprocessing steps, including tokenization and the removal of stopwords, the model is trained to predict subsequent words in a text sequence. The resulting model demonstrates an accuracy of 87%, showcasing its ability to generate coherent text based on the provided input context.**

I. Introduction

In this research, we developed a Long Short-Term Memory (LSTM) model to predict the next words in Shakespeare's plays, using a dataset consisting of 111,396 lines. For training purposes, we selected 3,000 lines and applied comprehensive preprocessing techniques. These steps included the removal of stop words, converting all words to lowercase to maintain uniformity, and tokenizing the text into individual words. Following this, n-gram sequences were created from the tokenized text, and padding was applied to ensure consistent input lengths for the model. The processed dataset was then divided into input features and labels, which were fed into a neural network architecture composed of an embedding layer, an LSTM layer, and a dense output layer. The LSTM model was trained to capture the sequential structure of the text and generate contextually accurate predictions based on the given input sequences. This approach aims to leverage the power of deep learning for text generation, providing insights into linguistic patterns in classical literature.

II. Ease of Use

For this Project, we created a user-friendly interface using Flask and HTML for next-word prediction, where users can type text and instantly receive predictions for the next few words based on their input. The interface allows users to decide how many words they want to predict, with the process happening in real-time. This setup offers a seamless experience, as predictions are generated dynamically as the user types, making it intuitive and efficient without the need for manual submission or page reloads. This real-time interaction enhances usability and provides a smooth, responsive user experience.

III. Prepare Your Paper Before Styling

In this section, we present some key points for understanding and applying the LSTM model for sentence completion.

A. Abbreviations and Acronyms

- **LSTM**: Long Short-Term Memory
- **RNN**: Recurrent Neural Network
- **DL**: Deep Learning
- **ML**: Machine Learning
- **UI**: User Interface

B. LSTM Equations

- **Forget Gate**:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

Decides what information to discard from the cell state.

- **Input Gate**:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

Determines which values to update in the cell state.

- **Cell State Update**:

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

Creates a vector of new candidate values to add to the cell state.

- **Cell State**:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Updates the cell state by forgetting the old state and adding new values.

- **Output Gate**:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

Decides what the next hidden state should be.

- **Hidden State**:

$$h_t = o_t * \tanh(C_t)$$

Produces the output of the LSTM cell.

C. Some Common Mistakes

Common mistakes in model development include:

- **Incorrect Hyperparameter Settings:** Setting hyperparameters improperly can significantly hinder model performance, leading to suboptimal results.
- **Inadequate Preprocessing:** Failure to preprocess the data correctly can negatively impact model effectiveness. This includes not removing stop words and punctuation, resulting in the retention of unnecessary data.

D. Authors and Affiliations

All people involved in the project are acknowledged appropriately.

ACKNOWLEDGMENT

The major headings are organized to offer clear insights into the construction of the LSTM model and the implementation of the user interface.

REFERENCES

[1] Ilaslan Düzgün, N. (2021). Next word prediction using LSTM with TensorFlow. *Medium Article*. https://medium.com/@ilaslanduzgun/next-wordprediction-using-lstm-with-tensorflow-e2a8f63b613c

[2] Kaggle. (n.d.). Shakespeare plays dataset. *Dataset*. https://www.kaggle.com/datasets/kingburrito666/shakespeareplays

[3] Parmar, K. (2021). Sentence autocomplete using TensorFlow. *Kaggle Code*. https://www.kaggle.com/code/kritikaparmargfg/sentenceautocomplete-using-tensorflow